

### REMARKS

Claims 1-31 and 33-44 are pending bust stand rejected. Claims 8-23 and 31 have been amended. Claim 32 has been cancelled. In view of the following remarks, the Applicant respectfully requests that the Examiner thoughtfully reconsider and pass the application on to issuance.

**SPECIFICATION OBJECTION:** Paragraph [0021] of the Specification has been amended to address the Examiner's concerns.

**SPECIFICATION OBJECTION:** The Examiner objected to the preambles of claims 24, 34, and 40. Each of these Claims recites a "computer-readable medium." The Examiner suggests that the Applicant replace this language with "A computer readable storage media" so that includes the language of paragraph [0021] of the Specification. The Examiner's attention is drawn to paragraph [0031] of the Specification which states that the disclosed object generator can be embodied in any "computer-readable medium." With this in mind, the preambles of Claims 24, 34, and 40 are consistent with the Specification and the Examiner's objection should be withdrawn.

**CLAIM REJECTIONS – 35 USC §101:** The Examiner rejected Claims 8-23 and 31-32 as being directed to non-statutory subject matter. These claims have been amended to address the Examiner's concerns.

**CLAIM REJECTIONS – 35 USC §103:** The Examiner rejected Claims 1-44 as being unpatentable over USPN 6,163,806 issued to Viswanathan in view of USPN 6,931,105 issued to Groves.

**Claim 1** is directed to a method for creating a best-match object at run time and recites the following:

1. receiving a request for an object;
2. polling object proxies for a confidence level representing the capability of each respective proxy to generate the requested object;
3. selecting one of the proxies based on the polled confidence level; and
4. directing the selected proxy to create the object.

Viswanathan discloses a system that provides access to “devices on a computer cluster.” Viswanathan, abstract. When a user requests that a particular device on the cluster be opened, an operating system kernel identifies information such as the device’s node and local address and then issues an “open request to the host node for the device.” Viswanathan, abstract. A special file system operating on the host node returns a “special file object that is associated with the desired device.” Viswanathan, abstract.

The following passage from Viswanathan explains further:

As described above, the kernel 242 handles system calls from the applications 150, such as requests to access the memory 230, the file system 206 or the devices 106. The kernel 242 differs from the kernel 132 (FIG. 1) as it has been modified by the present invention to support global device access. The proxy file system (PxFS) 244 is based on the Solaris PxFS file system but, like the kernel 242, is modified herein to support global device access. The PxFS 244 includes a collection of objects that enable an application 150-i in one node 202-i to interact seamlessly with the file system 206 across different nodes 202. The PxFS objects include PxFS clients 246, PxFS servers 248, f\_objs (file objects) 250, vnodes (virtual file nodes) 252, snodes (special file nodes) 254 and px\_vnodes (proxy vnodes) 256. Each of these objects is labeled in FIG. 6 as optional (opt) as they are created as needed by the PxFS 244 in response to operations of the file system 206.

Viswanathan, col. 9, lines 43-59. As a further example, Viswanathan explains:

An object such as the kernel 242 that needs to access the file system 206 first issues the access request to its local PxFS client 246. The PxFS client holds a reference to the PxFS server 248 co-located with the file system 206. This reference enables the PxFS client 246 to communicate the kernel's request to the file system 206 via the PxFS server 248. The file system 206 performs the requested access, creates a vnode object 252 representing the requested file and returns a reference to vnode object 252 to the PxFS server 248. Because the nodes 202-1 and 202-2 are different address spaces, the reference to the vnode 252 is useless to the PxFS client 246 and kernel 242 in the node 202-1. Consequently, the PxFS server 248 creates a file transport object (f\_\_obj) 250 linked to the vnode 252 and returns a reference to the f\_\_obj 150 to the PxFS client 246. Upon receiving the f\_\_obj reference the PxFS client 246 creates a proxy vnode (px\_\_vnode) 256 that is linked to the f\_\_obj 250. The kernel 242 can then access the file information represented by the vnode 252 by simply accessing the local px\_\_vnode 256.

Viswanathan, col. 16, lines 32-51.

Viswanathan teaches an operating system kernel that communicates with a proxy file system (PxFS) on behalf of an application. The proxy file system includes a collection of objects that allow the application to communicate with a remote file system. Those objects are created by the proxy file system as needed. The Kernel does not poll the proxy file system for a confidence level that represents the capabilities of each proxy that the proxy file system generates. Instead, Viswanathan's proxy file system simply generates each particular object as needed.

Consequently, Viswanathan fails to teach or suggest a method that includes polling object proxies for a confidence level representing the capability of each respective proxy to generate the requested object. Groves is silent on the matter. For at least these reasons, Claim 1 and Claims 2-7 which depend from Claim 1 are patentable over Viswanathan and Groves.

**Claim 8** recites a system that includes an object generator and a processor operable to execute the object generator. The object generator includes instructions that when executed by the processor function as:

1. means for receiving indicia of an object to be created;
2. means for identifying a select one of a plurality of object proxies responsive to a respective confidence level associated with each object proxy; and
3. means for directing the selected object proxy to create the object.

As discussed above Viswanathan and Groves fail to teach or suggest polling object proxies for a confidence level representing the capability of each respective proxy to generate the requested object. For the same reasons, those references fail to teach or suggest means for identifying a select one of a plurality of object proxies responsive to a respective confidence level associated with each object proxy. Viswanathan teaches a proxy file system that generates objects as needed. Viswanathan does not teach the section of that proxy file system “responsive to a respective confidence level associated with” the proxy file system. Groves is again silent on the matter.

For at least these reasons, Claim 8 and Claims 9-13 which depend from Claim 8 are patentable over Viswanathan and Groves.

**Claim 14** recites a system that includes an object generator and a processor operable to execute the object generator. The object generator includes instructions that when executed by the processor function as:

1. an object factory configured to poll object proxies capable of producing respective objects responsive to system needs; and
2. a pool including the object proxies for producing the object, the pool configured to receive indicia of the object from the object factory and each of the plurality of object proxies configured to return a respective confidence level responsive to the indicia.

As discussed above Viswanathan and Groves fail to teach or suggest polling object proxies for a confidence level representing the capability of each respective proxy to generate the requested object. For the same reasons, those references fail to teach or suggest “a pool including the object proxies . . . each of the plurality of object proxies configured to return a respective confidence level responsive to the indicia.”

Viswanathan teaches a proxy file system that generates objects as needed.

Viswanathan’s proxy file system is not a pool that includes a plurality of object proxies for producing the object where each of the plurality of object proxies is configured to return a respective confidence level responsive to the indicia. Groves is again silent on the matter.

For at least these reasons, Claim 14 and Claims 12-23 which depend from Claim 14 are patentable over Viswanathan and Groves.

**Claim 24** is directed to a computer readable medium that includes logic configured to implement the method of Claim 1. For at least the same reasons Claim 1 is patentable, so are Claim 24 and Claims 25-30 which depend from Claim 24.

**Claim 31** recites a system that includes an object generator and a processor operable to execute the object generator. The object generator includes instructions that when executed by the processor function as:

1. an object factory configured to receive a device identifier;
2. a pool having an interface configured to communicate with the object factory, the pool containing object proxies capable of producing respective objects; and
3. an object store coupled to the pool and configured to receive and retain objects generated by selected object proxies;
4. wherein the object factory is configured to poll a plurality of object proxies for a confidence level representing the capability of the respective object

proxy to generate an object suited for operating with a device responsive to the device identifier.

As discussed above Viswanathan and Groves fail to teach or suggest polling object proxies for a confidence level representing the capability of each respective proxy to generate the requested object. For the same reasons, those references fail to teach or suggest an object factory that is “configured to poll a plurality of object proxies for a confidence level representing the capability of the respective object proxy to generate an object suited for operating with a device responsive to the device identifier.” Viswanathan teaches a proxy file system that generates objects as needed. Viswanathan’s proxy file system is not polled for a confidence level representing the capability of the proxy file system to generate an object suited for operating with a device responsive to the device identifier. Groves is again silent on the matter.

For at least these reasons, Claim 31 is patentable over Viswanathan and Groves.

**Claim 33** is directed to a method for creating a best-match object at run time and recites the following:

1. loading a set of object proxies;
2. receiving indicia of a desired object for communicating with a peripheral device;
3. directing each of the object proxies to forward a confidence level representing the capability of each respective proxy to generate the desired object responsive to the indicia;
4. receiving a confidence level associated with an object proxy;
5. comparing the confidence level to a maximum confidence level, when the confidence level matches the maximum confidence level, directing the

- associated object proxy to generate an object, otherwise, recording the confidence level; and
6. determining if the confidence level exceeds the confidence level associated with a previously recorded confidence level, when the confidence level exceeds a previously recorded confidence level, recording an object proxy identifier, otherwise, determining if there are additional object proxies in the set, when there are additional object proxies, repeating the receiving a confidence level, comparing, and determining if the confidence level exceeds steps, otherwise, using the object proxy identifier to direct the associated object proxy to generate an object.

As discussed above Viswanathan and Groves fail to teach or suggest polling object proxies for a confidence level representing the capability of each respective proxy to generate the requested object. For the same reasons, those references fail to teach or suggest a method that includes (a) directing each of the object proxies to forward a confidence level representing the capability of each respective proxy to generate the desired object responsive to the indicia and (b) receiving a confidence level associated with an object proxy. Viswanathan teaches a proxy file system that generates objects as needed. Viswanathan's proxy file system is not directed to forward a confidence level representing the capability of the proxy file system to generate the desired object. Groves is again silent on the matter.

For at least these reasons, Claim 33 is patentable over Viswanathan and Groves.

**Claim 34** is directed to a computer readable medium that includes logic configured to implement the method of Claim 33. For at least the same reasons Claim 33 is patentable, so is Claim 34.

**Claim 35** is directed to a method for creating a best-match printer driver and recites the following:

1. receiving a request to use a printer;
2. polling printer driver proxies for a confidence level representing the capability of each respective printer driver proxy to generate a driver that when applied to data and forwarded to the printer will produce a useful representation of the data;
3. selecting one of the printer driver proxies based on the polled confidence level; and
4. directing the selected printer driver proxy to generate the driver.

As discussed above Viswanathan and Groves fail to teach or suggest polling object proxies for a confidence level representing the capability of each respective proxy to generate the requested object. For the same reasons, those references fail to teach or suggest a method that includes “polling printer driver proxies for a confidence level representing the capability of each respective printer driver proxy to generate a driver that when applied to data and forwarded to the printer will produce a useful representation of the data.” Furthermore, addressing Claim 35, the Examiner does not assert that Viswanathan or Groves mentions anything related to printer drivers let alone directing the selected printer driver proxy to generate the driver or polling printer driver proxies for information of any kind.

For at least these reasons, Claim 35 and Claims 36-39 which depend from Claim 34 are patentable over Viswanathan and Groves.

**Claim 40** is directed to a computer readable medium that includes logic configured to implement the method of Claim 35. For at least the same reasons Claim 35 is patentable, so is Claim 40 and Claims 41-44 which depend from Claim 40..



**CONCLUSION:** The foregoing is believed to be a complete response to the outstanding Office Action. Claims 1-31 and 33-44 are all felt to be in condition for allowance. Consequently, early and favorable action allowing these claims and passing the application to issue is earnestly solicited. The foregoing is believed to be a complete response to the outstanding Office Action.

Respectfully submitted,  
John L. Manuel

By /Jack H. McKinney/  
Jack H. McKinney  
Reg. No. 45,685

July 18, 2007